POCKET COMPUTER NEWSLETTER

© Copyright 1981 — Issue 08

THE SHAPE OF THE FUTURE?

The accompanying photograph shows a recently announced programmable calculator produced by Hewlett-Packard. While *PCN* likes to clearly delineate between "calculators" and "pocket computers" and normally does not report on the former, an exception is being made in this case. But, this exception is for a very particular reason.

As anyone who has followed Hewlett-Packard's line of calculators even on a casual basis can testify, the *shape* of the newly announced calculator is what is worthy of mention. It is radically different than the style(s) produced by the firm in the past. And, coincidently, it suddenly takes on a shape that will instantly be recognizable to most current readers of *PCN*.

The fact is, a number of industry speculators believe that world famous HP is shaping things up for a future entry into the pocket computer market. You need to get the shape of a PC into a basically horizontal format so that all of those necessary alphanumeric keys can be handled comfortably. And, it is a good idea to give the display some extra length over that generally allowed by the narrow width of the old style calculators.

It is even a good marketing idea to give customers some time to warm up to (and for the marketing department to "test the waters" around) a new product format before springing something as revolutionary as a *pocket computer* on the general public!

So there may be a good deal of truth to the rumors that these new HP calculators are in the general shape of what will eventually become full-fledged HP pocket computers.

How close to what *PCN* considers a true pocket computer is this new HP calculator? Real close. The unit features a liquid-crystal display. It has continuous memory that is retained even when the power (display) is turned off. It boasts of a number of "programming" capabilities, including the all important program "editing" functions. About all it needs to be classified as a pocket computer is a more sophisticated programming language, such as BASIC.

Indeed, the new programmable calculator has many features associated with currently available pocket computers such as the use of program labels which may be addressed by individual keys, the capability of using indirect addressing in a program, conditional tests on data and calculations, the availability of programming flags, nesting of subroutines up to four levels, and a memory extensive enough to permit some 200 "lines" (well, *short lines*) of program instructions. The editing features enable the insertion and deletion of individual program lines.

The calculator shown in the photograph, the HP-11C, comes with a full set of scientific functions that are key-selectable. In addition to commonly used math, engineering and statistics functions, there are special functions such as hyperbolics and a random-number generator.

The HP-11C is housed in a rugged case that is considered by Hewlett-Packard to be "slim line" and appropriate in size and weight for comfortable carrying in a shirt pocket.

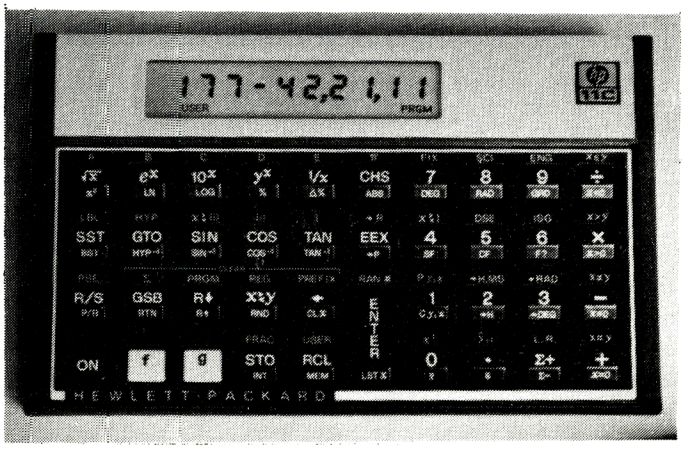
Another somewhat unique feature of the new unit is its ability to test itself. By pressing the appropriate keys, the user can direct the machine to conduct tests of its keyboard and calculating circuits! This capability can be comforting to the frustrated programmer who becomes convinced that the unit is malfunctioning! Incidently, the self-test capability does not interfere with data or programs stored in memory.

In addition to the HP-11C, which is primarily designed for scientists and engineers, there is a second unit, similar in appearance, that caters to financial analysts and businesspersons.

The HP-11C is reported to have a list price of $135.00 and the similar financial model (HP-12C) a slightly higher price of $150.00.

Will this be the shape of a *real* pocket computer produced by Hewlett-Packard? Look for the answer itself to appear during 1982!

Photo *Hewlett-Packard's HP-11C Shirt Pocket Calculator*



SUPER-WUMPUS

In 1973 Gregory Yob developed a computer game called *Wumpus.* The game quickly caught on and became one of the early classic games of personal computer users.

The objective in the game of Wumpus is to search a cave complex that contains twenty rooms connected by tunnels in order to locate and destroy a ferocious Wumpus. The cave complex contains two bats. Enter a cave containing a bat and you will be randomly relocated to another room. Two caves contain pits. Fall into a pit and it is all over! And, of course, there is the Wumpus. Meet the creature in a cave and it is all over for you.

When you are in a cave that is right next to a room containing any of the hazards, you will receive a warning message such as:

YOU FEEL A DRAFT.
YOU HEAR WINGS.
YOU SMELL A WUMPUS!

To destroy the Wumpus you use arrows which have the capability of darting through from one to five caves in a single shot.

The version of the game presented here for the pocket computer has several additional features that increase interest and difficulty.

First of all, you start with one to four arrows depending on which of twenty possible levels of play you select. In levels 1 through 19 you have a chance of finding new arrows when you have used your initial allotment.

Above level 2 earthquakes will move the pits about randomly to different caves. The warning message:

I FEEL AN EARTHQUAKE!

will let you know when a quake has occured.

If you try to shoot an arrow into a cave that is not connected by a tunnel or if you shoot into your own cave, you will, naturally, shoot yourself. What a way to go.

An arrow will only destroy the Wumpus if he is in the final cave of the arrow's flight path.

An arrow that passes through a cave in which the Wumpus is hiding, will quite likely cause the creature to shift caves. It thus may be wise to back up a room when you smell the creature and fire from a distance so that the Wumpus cannot move in on you.

Refer to the accompanying cave diagram to see how the caves are connected by tunnels. You may want to keep track of the progress of a game by drawing your own map as you proceed on a mission.

An accompanying table also shows the chances of finding another arrow when your initial supply is gone as well as the chance that an earthquake will occur, according to the level of play that has been selected.

Variables have been assigned so that those utilized the most have the fastest execution. All of memory is utilized so any user additions or changes will require deletions elsewhere in the program. A likely area for such modifications would be in the PRINT statements.

Line 90 will require a double entry operation in order to fit it all on one line unless you abbreviate statement terms such as GOSUB (use GOS.) and PAUSE (by using PA.). Line 175 also requires a double entry operation. To perform a double entry, first enter as much of the line as possible and press the ENTER key. Then use the cursor-right key to go to the end of the line and enter the rest of the line.

I execute most of my programs with the PC in the DEF mode. When in this mode, the program can be started by using SHIFT S. If you like to execute program with the PC in the RUN mode then you can eliminate the "S" label from line 35 and add the line:

1 GOTO 35

This modification adds a requirement for two steps of memory. To obtain the necessary space, I suggest you change the PRINT statement in line 85 to read:

"YOU FIND 1 ARROW."

Subroutines were placed at the beginning of the program. Line 5 contains a random number generating routine. You might want to use this routine in other programs. You can get any range of numbers desired for H by defining

H=(G*N/100)

whereby H will represent a random number having a value between 0 and N-1. Adding 1 to this value gives you a range of 1 to N.

Lines 10 through 30 determine the adjacent rooms.

Line 45 asks for the desired level of play and defaults to level 6.

Lines 50 through 75 set up the game and ensure that the player, the Wumpus, bats and pits are initialized to reside in separate caves.

Good luck and happy hunting!

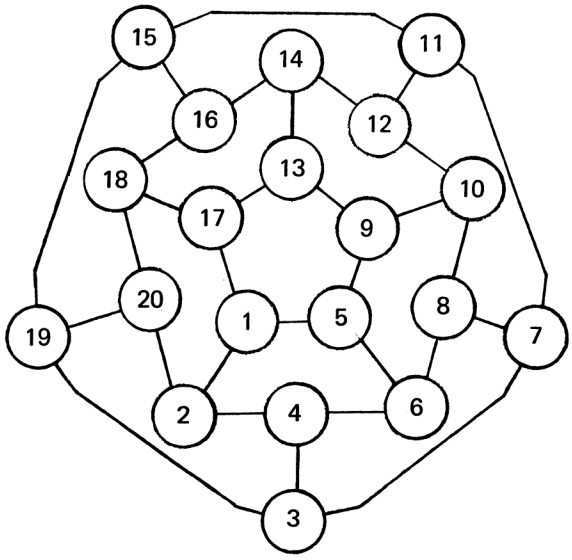
Super-Wumpus was submitted by: *Gary Heidbrink, Box 24, FPO New York, NY 09593.*

[P.S. I think Mr. Heidbrink has done a super job of programming this game on a PC! — N.W.]

VARIABLES USAGE TABLE

A(1)	First cave being shot
A(2)	Second cave being shot
A(3)	Third cave being shot
A(4)	Fourth cave being shot
A(5)	Fifth cave being shot
F	Used in random number generator, prevents cycling
G	Main random number routine variable
H	Generated random number
I	Number of arrows
J	Number of turns taken
K	Prevents bats from taking you after dropping you off
L$	Holds responses from input statements
M	Location of the Wumpus
N	First pit location
O	Second pit location
P	First bat location
Q	Second bat location
R	Data variable
S	Number of caves being shot
T	FOR/NEXT variable
U	Data variable
V	Level of difficulty
W	Current room being analyzed
X	First adjacent room
Y	Second adjacent room
Z	Third adjacent room

```
 5:F=F+1:G=4391
   47*6+F:G=23*
   G-INT (G*.23
   )*100:H=INT
   (G*.2)+1:
   RETURN
10:X=W:Y=W:Z=W:
   IF W/2<>INT
   (W/2)LET X=X
   +3:GOSUB 15
15:X=X-1:IF X<1
   LET X=X+20
20:Y=Y+2:IF Y>2
   OLET Y=Y-20
25:Z=Z-2:IF Z<1
   LET Z=Z+20
30:RETURN
35:"S":F=1:G=32
   54:INPUT "RN
   D #: ";G
40:PRINT "
    SUPER-WUMPU
   S"
45:V=6:INPUT "L
   EVEL (1-20)
   ";V:IF (V<1)
   +(V)20)GOTO
   45
50:J=0:K=0:I=4-
   INT (V/6):
   GOSUB 5:M=H
55:GOSUB 5:N=H:
   IF N=MGOTO 5
   5
60:GOSUB 5:O=H:
   IF (O=M)+(O=
   N)GOTO 60
65:GOSUB 5:P=H:
   IF (P=M)+(P=
   N)+(P=O)GOTO
   65
70:GOSUB 5:Q=H:
   IF (Q=M)+(Q=
   N)+(Q=O)+(Q=
   P)GOTO 70
75:GOSUB 5:W=H:
   IF (W=M)+(W=
   N)+(W=O)+(W=
   P)+(W=Q)GOTO
   75
80:BEEP 1:PAUSE
   "YOU ARE IN
   ROOM ";W
85:GOSUB 5:IF (
   I=0)IF (H>V)
   LET I=1:BEEP
   1:PAUSE "YOU
    FOUND AN AR
   ROW."
90:IF H<V/2BEEP
   2:PAUSE "YOU
    FEEL AN EAR
   THQUAKE!":
   GOSUB 5:R=N:
   N=H:IF F/2=
   INT (F/2)LET
   O=R
95:IF W=MBEEP 4
   :PAUSE "WUMP
   US ATE YOU."
   GOTO 40
100:IF (W=N)+(W=
   O)BEEP 4:
   PRINT "YOU F
   ELL INTO A P
   IT.":GOTO 40
105:IF K<>0GOTO 120
110:IF W=PLET P=
   Q:Q=W
115:IF N=QBEEP 2
   :PAUSE "BAT
   GRABBED YOU.
   ":K=1:GOSUB
   5:Q=H:W=H:
   GOTO 80
120:GOSUB 10
125:IF (X=M)+(Y=
   M)+(Z=M)BEEP
   3:PAUSE "YOU
    SMELL A WUM
   PUS!"
130:IF (X=N)+(Y=
   N)+(Z=N)+(X=
   O)+(Y=O)+(Z=
   O)BEEP 1:
   PAUSE "YOU F
   EEL A DRAFT.
   "
135:IF (X=P)+(Y=
   P)+(Z=P)+(X=
   Q)+(Y=Q)+(Z=
   Q)BEEP 1:
   PAUSE "YOU H
   EAR WINGS."
140:PRINT "TUNNE
   LS GO TO:";
   USING "###";
   X;Y;Z
145:INPUT "MOVE
   OR SHOOT? (M
   /S): ";L$:J=
   J+1:IF L$="M
   "GOTO 160
150:IF L$="S"
   GOTO 170
155:J=J-1:GOTO 1
   40
160:INPUT "ENTER
   ING WHICH CA
   VE? ";H:IF (
   H=X)+(H=Y)+(
   H=Z)LET W=H:
   K=0:GOTO 80
165:PAUSE "YOU C
   ANNOT GO THE
   RE.":GOTO 15
   5
170:IF I=0PAUSE
   "SORRY, NO A
   RROWS.":GOTO
   155
175:INPUT "THRU
   HOW MANY CAV
   ES? ";S:IF (
   S>5)+(S<1)
   PRINT "CAN S
   HOOT TO 1-5
   CAVES.":GOTO
   155
180:I=I-1:FOR T=
   1TO S:INPUT
   "THRU WHICH
   CAVE? ";R:A(
   T)=R:NEXT T
185:R=W:FOR T=1
   TO S:W=A(T):
   IF R<>WIF (X
   =W)+(Y=W)+(Z
   =W)LET W=W:
   GOSUB 10:
   NEXT T:GOTO
   195
190:BEEP 4:PRINT
   "MISFIRE! SH
   OT SELF.":
   GOTO 40
195:IF N=MBEEP 5
   :PRINT "SHOT
    WUMPUS IN";
   J;" TURNS.":
   GOTO 40
200:N=M:GOSUB 10
   :GOSUB 5:T=2
   3+INT (H/4):
   M=A(T)
205:N=R:GOTO 80
```



LEVEL OF PLAY	STARTING ARROWS	CHANCE TO FIND ARROW	CHANCE OF QUAKE
1	4	95%	--
2	4	90%	--
3	4	85%	05%
4	4	80%	05%
5	4	75%	10%
6	3	70%	10%
7	3	65%	15%
8	3	60%	15%
9	3	55%	20%
10	3	50%	20%
11	3	45%	25%
12	2	40%	25%
13	2	35%	30%
14	2	30%	30%
15	2	25%	35%
16	2	20%	35%
17	2	15%	40%
18	1	10%	40%
19	1	05%	45%
20	1	--	45%

PLOTTER PROGRAM

The program shown in the accompanying listing plots a graph of whatever function is specified in line 50. Note that in line 50, Y is calculated based on the value of X.

The program requests the minimum and maximum values of X and Y that are to be plotted as well as the number of intervals to be used. The values of X used in the program are determined by subdividing the interval between MIN X and MAX X as specified.

The graph is printed lengthwise along the tape. Minimum and maximum values of X and Y are printed at the start of the graph.

If a plotted point lies outside the specified range (because Y goes beyond the MIN or MAX indicated), a blank line is printed in order to maintain proper spacing.

Some examples that give interesting results include:

Sine curve (illustrated)
Line 50: Y=SIN X (DEGree mode!)
MIN X=0; MAX X=360; MIN Y=-1; MAX Y=1; 15 intervals

Normal probability curve
Line 50: Y=EXP(-XX/2)/$\sqrt{2\pi}$
MIN X=-3; MAX X=3; MIN Y=0; MAX Y=.4; 16 intervals

Parabola
Line 50: Y=XX
MIN X=-2; MAX X=2; MIN Y=0; MAX Y=4; 8 intervals

Damped harmonic motion
Line 50: Y=EXP-.2X*COS X (RADian mode!)
MIN X=0; MAX X=12; MIN Y=-1; MAX Y=1; 12 intervals

Semicircle (appears as a semi-ellipse)
Line 50: Y=$\sqrt{(1-XX)}$
MIN X=-1; MAX X=1; MIN Y=0; MAX Y=1; 12 intervals

Wave with harmonic
Line 50: Y=SIN X + SIN 2X (DEGree mode)
MIN X=0; MAX X=360; MIN Y=-1.75; MAX Y=1.75;
12 intervals

Program submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

Example *Sine curve*

```
MIN X=0.
MAX X=360.
MIN Y=-1.
MAX Y=1.

                *
                  *
                    *
                      *
                      *
                    *
                  *
                *
              *
            *
          *
        *
      *
      *
        *
          *
            *
```

Program *Plotter*

```
10:INPUT "MIN X
   ?";A,"MAX X?
   ";B,"MIN Y?"
   ;C,"MAX Y?";
   D,"NO OF INT
   ERVALS?";N
15:PRINT "MIN X
   =";A:PRINT "
   MAX X=";B:
   PRINT "MIN Y
   =";C:PRINT "
   MAX Y=";D:
   PRINT " "
20:FOR Z=0TO N:
   X=A+Z/N*(B-
   A
50:Y=SIN X
75:P=INT (1.5+1
   5*(Y-C)/(D-C
   :GOSUB 80:
   NEXT Z:END
80:R$="      ":
   S$=R$:Q=INT
   .16P:IF PIF
   P<17GOTO 82+
   Q
81:PRINT R$:
   RETURN
82:R=0
83:S=0
84:GOTO 84+P-6Q
85:PRINT R$;S$;
   "*":RETURN
86:PRINT R$;S$;
   " *":RETURN
87:PRINT R$;S$;
   "  *":RETURN
88:PRINT R$;S$;
   "   *":
   RETURN
89:PRINT R$;S$;
   "    *":
   RETURN
90:PRINT R$;S$;
   "     *":
   RETURN
```

MULTIPLE-STRIP PLOTTER PROGRAM

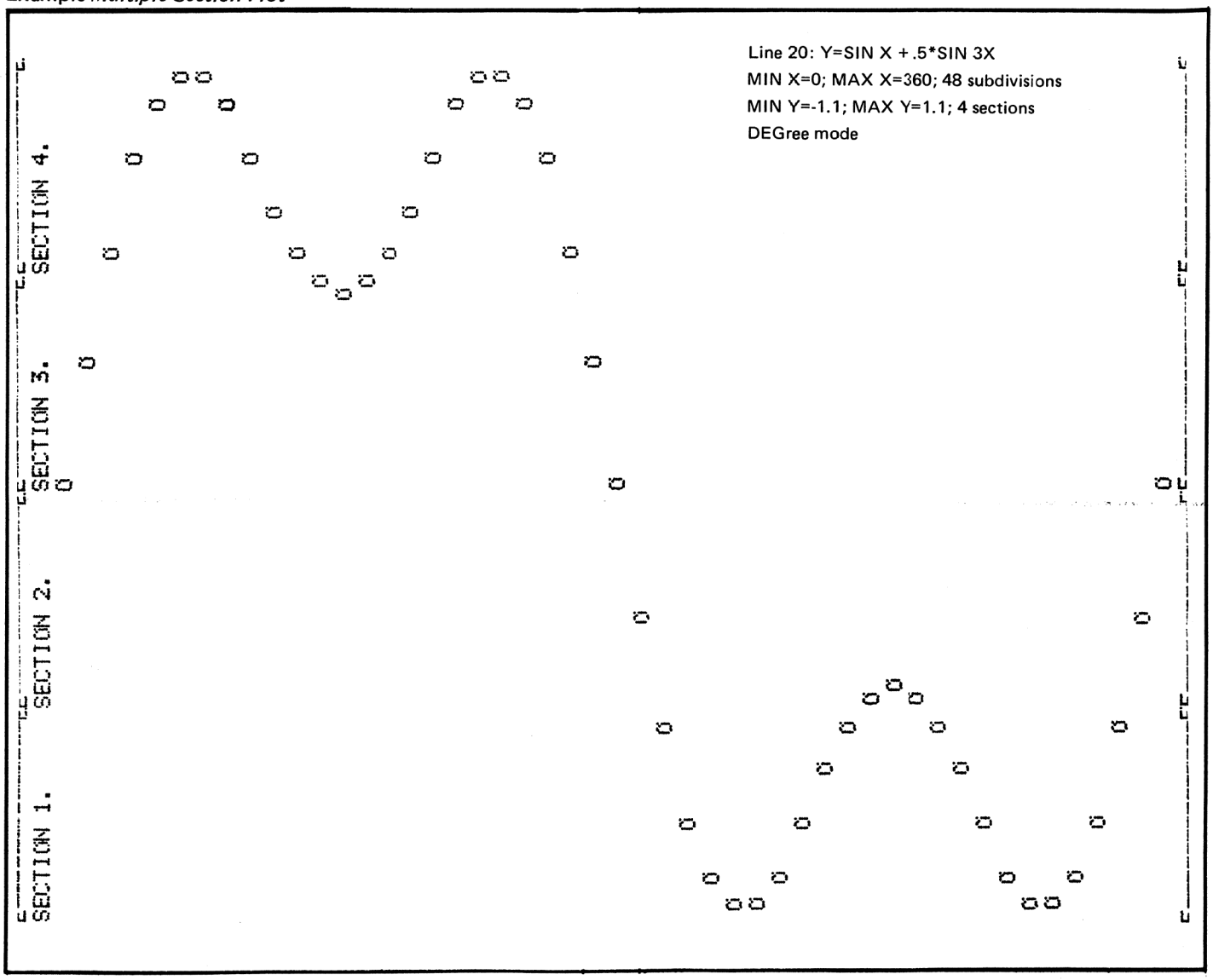
This program produces a larger graph than the previously described plotter program by utilizing several strips of paper. The number of strips or sections that may be utilized are specified by the user. After the individual sections have been plotted, the left edge of each section (except the first) can be trimmed. Then the sections are pasted alongside one another. The program provides markings on the strips to aid in cutting and

A spray adhesive such as *Scotch Spray Mount* (catalog number 6065) considerably simplifies the task of mounting the strips.

Program *Multiple-Strip Plotter*

```
10:INPUT "MIN X
   ?";A,"MAX X?
   ";B,"NO OF S
   UBDIVISIONS?
   ";N,"MIN Y";
   C,"MAX Y";D
11:INPUT "NO OF
    SECTIONS?";
   M:FOR J=1TO
   M+1:PRINT "E
   ----------
   --E":IF J>M
   END
12:PRINT "SECTI
   ON ";J:FOR I
   =0TO N:X=A+(
   B-A)I/N
85:P=INT (17.5-
   16J+(16M-1)*
   (Y-C)/(D-C:
   GOSUB 90:
   NEXT I:NEXT
   J
90:Q$="      ":
   R$=Q$:S=INT
   .16P:IF PIF
   P<17GOTO 92+
   S
91:PRINT Q$:
   RETURN
92:Q=0
93:R=0
94:GOSUB 94+P-6
   S:PRINT Q$;R
   $;S$;"O":
   RETURN
95:S=0:RETURN
96:S$=" ":
   RETURN
97:S$="  ":
   RETURN
98:S$="   ":
   RETURN
99:S$="    ":
   RETURN
100:S$="     ":
   RETURN
```

Example *Multiple-Section Plot*



The user must specify a routine for the calculation of the variable Y as a function of X in lines 20 through 84.

The symbol used for plotting is specified in line 94 of the program. The letter O is used in the example.

A graph measuring approximately 5.5 by 7 inches in size is created, for example, by specifying 4 sections and using 48 subdivisions.

This program is a *Norlin Rober* creation.

STRING PRINTER

To use your CE-122 or Radio Shack PC Printer as a message unit, load this program into your pocket computer:

```
100 CLEAR:A=2
110 INPUT "STRING? ";A$(A):A=A+1:GOTO 110
120 PRINT B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;M$;N$
130 GOTO 100
```

Execute RUN. Key in up to 7 characters, press ENTER, repeat as required. Pressing ENTER with no other input causes the message to be printed. Thanks to: *David Motto, 3639 Roosevelt Circle, Jackson, MI 49203.*

PROGRAM CALCULATES REMAINING PRINTER TAPE

Hank Librach, 52 Bulkley Drive, Fairfield, CT 06430, sent in this program that assumes a paper thickness of 0.003 and an inside diameter of 0.5 inches. How much is left on your roll?

Program *Tape Length Calculator*

```
10:PRINT "TAPE
   LENGTH"
15:PRINT "CALCU
   LATOR"
16:PRINT " "
20:PRINT "MEASU
   RE SPOOL"
25:PRINT "OUTSI
   DE DIAMETER"
30:INPUT "DIAM=
   ";D
35:PRINT "DIAM.
   = ";D
40:A=(D^2-.25)*
   π/4
50:L=A/.003
51:PRINT " "
60:PRINT "REMAI
   NING LENGTH=
   ";L;" IN."
```

```
TAPE LENGTH
CALCULATOR

MEASURE SPOOL
OUTSIDE DIAMETER
DIAM.= 0.75

REMAINING LENGTH
=81.8123087 IN.
```

AN ASSORTMENT OF SORTS

Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158, sent in the following assortment of sort routines.

Given a list of N numbers, with N greater than 2, stored in array locations A(27) to A(26+N), the sorting routines shown here rearrange the N numbers in order from smallest to largest. Although more than 160 numbers can be sorted, it should be noted that sorting a long list on a PC takes a long, *long* time. In the routines illustrated, the variable named "N" should contain the number of items to be sorted.

The Bubble Sort

Starting from the beginning of the list, adjacent pairs of numbers are compared; if out of order, they are interchanged. Thus, the larger number is placed lower in the list. The process is repeated until the list is completely ordered. Altogether, (N)*(N-1)/2 comparisons are required.

```
10 FOR X=25+N TO 27 STEP -1:FOR Y=27 TO X
20 IF A(Y)>A(Y+1) LET Z=A(Y):A(Y)=A(Y+1):A(Y+1)=Z
30 NEXT Y:NEXT X
```

The routine uses 75 bytes and variables N, X, Y and Z.

The Exchange Sort

The smallest number in the list is placed into Z, with its location in W; it is then exchanged with the first number in the list. Next, the second smallest number is found and exchanged with the second number in the list; and so on. Although this requires as many comparisons as the bubble sort, fewer stores and recalls are needed, giving a somewhat shorter execution time.

```
10 FOR X=27 TO 25+N:W=X:Z=A(X):FOR Y=X+1 TO
   26+N
20 IF A(Y)<Z LET W=Y:Z=A(Y)
30 NEXT Y:A(W)=A(X):A(X)=Z:NEXT X
```

The routine uses 82 bytes and variables N, W, X, Y and Z.

The Insertion Sort

The list is treated as if it is divided into a sorted part and an unsorted part. Numbers are removed from the unsorted part and inserted into the sorted part, one at a time. A space for the insertion is provided by shifting down all the numbers in the sorted part that are below the insertion position.

If the list starts out completely reversed, (N+2)*(N-1)/2 comparisions are needed. Fewer comparisons are required if the list is initially partially ordered. This sort performs half as many stores and recalls as that required by the exchange sort.

```
10 FOR X=28 TO 26+N:Z=A(X):FOR Y=X-1 TO 26
   STEP -1
20 IF Z<A(Y) LET A(Y+1)=A(Y):NEXT Y
30 A(Y+1)=Z:NEXT X
```

The routine uses 72 bytes and variables N, X Y and Z.

The Shell Sort

A Shell sort is more complex than the methods described previously. Subsets of the list, consisting of the numbers whose locations are separated by multiples of V, ares sorted using an insertion technique. The values of V utilized in the routine illustrated are 121, 40, 13, 4 and 1; starting with a value that is than N. (The initial value for V is determined in line 10 of the program.) In comparison with the other routines presented, there is typically a considerable reduction in the number of comparisons required to accomplish a sort, as well as a reduction in the number of stores and recalls executed.

```
10 V=INT(.5*3ΔINT(LOG 2N/LOG 3
20 FOR W=27 TO 26+V:FOR X=W+V TO 27+N-V
   STEP V:Z=A(X):FOR Y=X-V TO W STEP -V
30 IF Z<A(Y) LET A(V+Y)=A(Y):A(Y)=Z:NEXT Y
40 NEXT X:NEXT W:V=(V-1)/3:IF V GOTO 20
```

The routine uses 124 bytes and the variables N, V, W, X, Y and Z.

Comparisons

Average times for sorting N randomly arranged numbers are tabulated in the accompanying table. The time is noted in minutes and seconds. With the exception of the Shell sort, execution time is approximately proportionate to N^2

Table *Timing Comparisons*

N	Bubble	Exchange	Insertion	Shell
10	0:37	0:28	0:22	0:26
20	2:30	1:40	1:15	1:20
30	5:35	3:35	3:05	2:05
40	10:00	6:10	4:35	2:55
50	15:40	9:30	7:00	4:40

It may be noted that the speed of the Shell sort becomes especially significant as a list lengthens.

The next table tabulates the time required to sort a list of 50 items when it is initially ordered versus being initially completely reversed.

Table *Ordered and Reversed Lists*

	Bubble	Exchange	Insertion	Shell
In order:	10:25	8:35	0:50	2:45
Reversed	20:50	10:25	13:05	3:55

Note, in the case of the Shell sort, that an initially reversed list does not necessarily require more time than when the list is in some other (less than ordered) arrangement. In the other types of sorts, the completely reversed list requires the most amount of execution time.

It is also interesting to note that the insertion type of sort is the fastest when a list is initially ordered or is close to that state.

ROOTS OF POLYNOMIALS

The accompanying program, provided by *Norlin Rober,* determines the complex roots of a polynomial having real coefficients, with degree 2 through 5.

To use the program, execute RUN and enter the coefficients as prompted, according to the following format:

Degree 2: $Ax^2+Bx+C=0$, with $A \neq 0$

Degree 3: $Ax^3+Bx^2+Cx+D=0$, with $A \neq 0$

Program *Polynomials*

```
10:INPUT "DEGRE
   E? ";N:GOTO
   10N
20:INPUT "A=";A
   ,"B=";B,"C="
   ;C
21:IF 4AC-BBLET
   X=-B/2A:Y=√(
   4AC-BB)/2A:
   PRINT "RE(X)
   =";X:PRINT "
   IM(X)=";Y:
   END
22:X=(-B+√(BB-4
   AC))/2A:Y=(-
   B-√(BB-4AC))
   /2A:PRINT "X
   =";X:PRINT "
   X=";Y:END
30:INPUT "A=";A
   ,"B=";B,"C="
   ;C,"D=";D
31:G=BB-3AC:H=4
   .5A*(BC-3AD)
   -BBB:K=HH-GG
   G
32:IF KLET M=
   SGN H*(G/(√K
   +ABS H)^(1/3
   )+(√K+ABS H)
   ^(1/3:Y=(-M-
   2B)/6A:Z=√(3
   MM-12G)/ABS
   6A
33:IF K=0LET M=
   √4G*SGN H:Y=
   (-√G*SGN H-B
   )/3A:Z=Y
34:IF -KLET M=√
   4G*COS (ACS
   (H/√GGG)/3:Y
   =(√(12G-3MM)
   -M-2B)/6A:Z=
   (-√(12G-3MM)
   -M-2B)/6A
35:X=(M-B)/3A:
   PRINT "X=";X
   :IF KPRINT "
   RE(X)=";Y:
   PRINT "IM(X)
   =";Z:END
36:PRINT "X=";Y
   :PRINT "X=";
   Z:END
40:INPUT "A=";A
   ,"B=";B,"C="
   ;C,"D=";D,"E
   =";E:IF -A
   LET A=-A:B=-
   B:C=-C:D=-D:
   E=-E
41:G=CC+12AE-3B
   D:H=CCC+4.5*
   (2ADD-8ACE+3
   BBE-BCD:K=HH
   -GGG:IF K=0
   LET M=√G*(1+
   (H)0
42:IF KLET M=
   SGN H*(G/(√K
   +ABS H)^(1/3
   )+(√K+ABS H)
   ^(1/3
43:IF -KLET M=√
   4G*COS (ACS
   (H/√GGG)/3
44:Q=9BB+12A*(M
   -2C:IF -QLET
   Q=0
45:R=0:IF (C+M)
   *(C+M)/9-4AE
   LET R=√((C+M
   )*(C+M)/9-4A
   E:IF 6AD-BC-
   RMLET R=-R
46:FOR S=-1TO 1
   STEP 2:T=6B*
   (3B-S*√Q)+12
   A*(6RS-M-4C
47:IF -TLET X=(
   S*√Q-3B)/12A
   :Y=√-T/12A:
   PRINT "RE(X)
   =";X:PRINT "
   IM(X)=";Y:
   NEXT S:END
48:X=(S*√Q-3B+√
   T)/12A:Y=(S*
   √Q-3B-√T)/12
   A:PRINT "X="
   ;X:PRINT "X=
   ";Y:NEXT S:
   END
50:INPUT "A=";A
   ,"B=";B,"C="
   ;C,"D=";D,"E
   =";E,"F=";F:
   IF -ALET A=-
   A:B=-B:C=-C:
   D=-D:E=-E:F=
   -F
51:X=-SGN F
52:P=(((AX+B)X
   +C)X+D)X+E)X
   +F:IF FPLET
   X=7X:GOTO 52
53:IF ABS PLET
   W=SGN X*INT
   (ABS .5X:Y=X:
   X=.5*(W+Y
54:IF ABS PLET
   P=(((AX+B)X
   +C)X+D)X+E)X
   +F:A(24-SGN
   FF):X:X=.5*(
   W+Y:IF X<>Y
   GOTO 54
55:PRINT "X=";X
   :B=(((AX+B)X
   +C)X+D)X+E:D
   =((AX+B)X+C)
   X+D:C=(AX+B)
   X+C:B=AX+B:
   GOTO 41
```

Degree 4: $Ax^4+Bx^3+Cx^2+Dx+E=0$, with $A \neq 0$

Degree 5: $Ax^5+Bx^4+Cx^3+Dx^2+Ex+F=0$, with $A \neq 0$

All complex roots will be calculated and displayed. If there is a conjugate pair of complex roots, the real part will be indicated as RE(X) and the imaginary part as IM(X).

A loss of accuracy may occur, particularly in the case of multiple roots. This is unavoidable with floating-point arithmetic.

An example: To solve the equation:

$$8x^5-75x^4+275x^3-485x^2+387x-90=0$$

type RUN and press ENTER. In response to the DEGREE? query, input 5. Enter the coefficients as prompted. In this case A=8, B=-75, C=275, D=-485, E=387 and F=-90. Solutions for this problem will be displayed as:

X = 0.375
RE(X) =2 indicating the conju-
IM(X) = 1 gate pair: $2 \pm i$
X = 3
X = 2

Program Notes

Second-degree equations are solved by lines 20 – 22; third-degree by lines 30 – 36; fourth-degree, lines 40 – 48; and fifth-degree by lines 50 – 55 (using half-interval search) with reduction to 4'th degree in line 55, with the resulting quartic equation being solved beginning at line 41.

Determination of the first root of a 5'th degree equation may take up to one minute.

The cubic equation is solved by the use of Cardan's Formulas; the quartic is solved by a modification of Ferrari's method.

PROGRAM CONVERTS FRACTIONS & DECIMALS

The program provided here as submitted by *Norlin Rober,* should be operated with the PC in the DEF mode.

SHIFT F is used to obtain a fraction approximating a given decimal. The input must be a positive number with a non-zero fractional part. A sequence of increasingly accurate approximations is provided. Each approximating fraction given is reduced to lowest terms.

Example: Push SHIFT F and use 3.141592654 (π) as input, then press ENTER. The program responds with the approximations:

3.142857143 = 22/7
3.141509434 = 333/106
3.14159292 = 355/113
3.141592654 = 10438/33215

The latter result agrees to 10 significant digits.

SHIFT D produces the decimal value for a given fraction. The fraction is reduced to lowest terms and displayed, then the infinite decimal expansion is determined. Both the numerator and denominator inputted must be positive.

Example: Push SHIFT D, respond with 735 ENTER to the NUM? query and 456 ENTER to the DEN? query. The fraction will initially be reduced and shown as: 2 plus 37/104. Hit the ENTER key again to compute the infinite decimal. The computer displays, for this case, 3 places of non-repeating

Program *Fractions & Decimals*

```
10:"F"N=0:D=1:A
   =1:B=1:C=0:
   INPUT "DECIM
   AL? ";X:F=X
11:E=INT (A/F:G
   =F:F=A-EG:A=
   G:G=N:N=B+EG
   :B=G:G=D:D=C
   +EG:C=G:IF D
   <2GOTO 11
12:G=N/D:PRINT
   G:PRINT " ="
   ;USING "####
   ########";N;"
   /";USING ;D:
   IF G<>XGOTO
   11
13:PRINT " AGRE
   ES TO 10 DIG
   ITS":GOTO 12
20:"I"INPUT "NU
   M? ";N,"DEN?
    ";D:C=INT (
   N/D:N=N-CD:A
   =N:B=D
21:E=B:B=A-B*
   INT (A/B:A=E
   :IF BGOTO 21
22:N=N/E:D=D/E:
   A=D:F=0
23:PRINT "INTEG
   RAL PART; ";
   C:PAUSE "FRA
   C PART (REPU
   TED)=":PRINT
   USING "######
   ######";N;"/
   ";USING ;D
24:IF .5A=INT .
   5ALET A=.5A:
   B=B+1:GOTO 2
   4
25:IF .2A=INT .
   2ALET A=.2A:
   F=F+1:GOTO 2
   5
26:C=1:E=0:IF F
   >BLET B=F
27:C=10C-A*INT
   (10C/A:E=E+1
   :IF C>1GOTO
   27
28:C=E4N:BEEP 1
   :PAUSE "NO.
   OF DECIMAL P
   LACES="
29:PRINT USING
   "#####";B;"
   (NON-REPEATI
   NG)":PRINT
   E;" (REPEATI
   NG BLOCK).":
   USING
30:A=INT (C/D:
   PRINT "1ST 4
    PLACES; ";A
31:A=.01A:C=E4C
   -E4D*INT A-E
   4D*(A-INT A:
   A=INT (C/D:
   PRINT "NEXT
   4; ";A:GOTO
   31
```

digits, to be followed by a repeating block of digits. Use the ENTER key repeatedly to obtain digits (in blocks of four):

.611842105263157894736

Program Notes

In line 11, approximating fractions are determined using the convergents of a continued fraction.

In lines 24 & 25, the length of the non-repeating block of decimal places is determined by counting the 2's and 5's that are removed as factors of the denominator. The larger of the two counts is the number of digits in the non-repeating block.

In line 27, a count is made of the number of divisions required before the remainder repeats itself. The count indicates the size of the repeating block.

Counting the number of divisions in line 27 may take a long time. A denominator of 1087, for example, results in a repeating block of 1086 digits, which takes almost 12 minutes to be determined. If the block size should happen to be a 10-digit number, execution time would be approximately 200 years. (Be sure to start with a fresh set of batteries for this case!)

P.O. Box 232, Seymour, CT 06483

FROM THE WATCH POCKET

Look for the Panasonic/Quasar HHC to be announced with a lot of fanfare around the end of October.

Meanwhile, the Casio FX-702P unit is being delivered and snapped up. A check with *Markline, Inc., P.O. Box 6800, Waltham, MA 02154,* a distributor of the unit, indicated they quickly sold out their initial supply. They expect more shortly. You can check their availability of the unit by phoning (617) 891-6250.

If you need any Sharp products, including supplies such as paper and ribbons, check with *Atlantic N.E. Marketing, P.O. Box 921, Marblehead, MA 01945.* The phone number is (617) 639-0285.

Sharp is currently selling the PC-1211 at $179.95. However, if you wait a few months you are likely to see Radio Shack make a move for the holiday market at around $169.95.

Bob McElwain continues to do a nice job with his column in **Interface Age.** In the October issue he discusses an application of interest to educators.

George Haller has an automobile trip calculator program designed to run on a Radio Shack TRS-80/Sharp PC-1211 in the September/October issue of **80-U.S. Journal.**

Page 162 of the September, 1981, issue of **BYTE** contains a programming tip of interest to PC users. It explains how to convert a FOR...NEXT loop in the BASIC language to the REPEAT...UNTIL construct that is desirable in some types of applications. The method discussed is easily implemented on a PC so you might want to look it over if you get a chance.

Don't forget to send in your entries for the *Hot Steaming Contest* as described in last month's *PCN.* Submissions must be received by October 20, 1981, in order to be eligible for consideration by the judges. We already have a number of clever messages entered. Don't miss out. Remember, the prizes are *cash* awards!

It is well over a year since I obtained my first PC and I have yet to change the batteries. — *Nat Wadsworth, Editor*

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others? Have you developed application programs that you would like to share with fellow pocket computer enthusiasts? Have you had experience with PC products or programs that others might find of value?

If so, write it up! If your submission is accepted for publication you will earn a cash reward as well as have the satisfaction of knowing you are helping other PC users.

We prefer that program listings be supplied on audio cassette tape so that they may be easily reviewed. We can also then make listings of accepted programs directly from the tape to avoid transcription errors. If we cannot use your material, we will endeavor to promptly return it *provided you include a self-addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483